دانشکده مهندســی

پایان‌نامه کارشناسی ارشد در رشته مهندسی کامپیوتر (نرم افزار)

ارائه روشی نوین برای خطازدایی خودکار نرم‌افزار

توسط:

حسن طنابی

استاد راهنما:

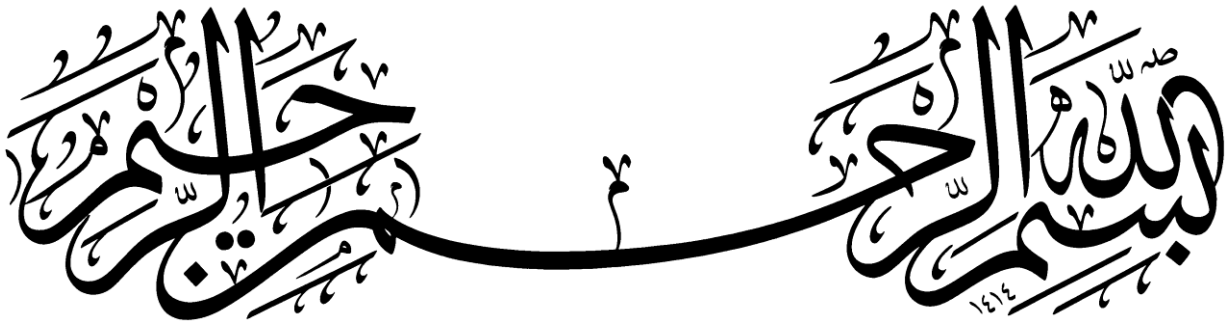
دکتر اشکان سامی

استادان مشاور:

دکتر محمدهادی صدرالدینی

دکتر غلامحسین دستغیبی فرد

شهریور ۱۳۹۱


بسم الله الرحمن الرحيم
١٤١٤

به نام خدا

ارائه روشی نوین برای خطازدایی خودکار نرم‌افزار

به وسیله‌ی:

حسن طنابی

پایان نامه

ارائه شده به تحصیلات تکمیلی دانشگاه به عنوان بخشی

از فعالیت‌های تحصیلی لازم برای اخذ درجه کارشناسی ارشد

در رشته:

مهندسی کامپیوتر- نرم افزار

از دانشگاه شیراز

شیراز

جمهوری اسلامی ایران

ارزیابی شده توسط کمیته پایان نامه با درجه:

دکتر اشکان سامی، استادیار بخش مهندسی و علوم کامپیوتر .....

دکتر محمد هادی صدرالدینی، دانشیار بخش مهندسی و علوم کامپیوتر .....

دکتر غلامحسین دستغیبی فرد، استادیار بخش مهندسی و علوم کامپیوتر و فناوری اطلاعات (رئیس کمیته) .....

شهریورماه ۱۳۹۱

به نام خدا

اظهارنامه

اینجانب حسن طنابی (۸۹۰۲۱۹) دانشجوی رشته‌ی مهندسی کامپیوتر گرایش نرم افزار دانشکده‌ی مهندسی اظهار می‌کنم که این پایان‌نامه حاصل پژوهش خودم بوده و در جاهایی که از منابع دیگران استفاده کرده‌ام، نشانی دقیق و مشخصات کامل آن را نوشته‌ام. همچنین اظهار می‌کنم که تحقیق و موضوع پایان‌نامه‌ام تکراری نیست و تعهد می‌نمایم که بدون مجوز دانشگاه دستاوردهای آن را منتشر ننموده و یا در اختیار غیر قرار ندهم. کلیه حقوق این اثر مطابق با آیین‌نامه مالکیت فکری و معنوی متعلق به دانشگاه شیراز است.

تقدیم به

پدر و مادر عزیزم

به پاس تعبیر عظیم و انسانی‌شان از کلمه ایثار،
به پاس عاطفه سرشار و گرمای امیدبخش وجودشان،
به پاس قلب‌های بزرگشان که فریاد رس روزهای سخت من بودند،
و به پاس تمام محبتهای ایشان که هیچگاه نمی توانم جبران کنم....

و سپا سگزارم از

تمامی اساتیدی که در این مدت با راهنمائی ها و نظرات سازنده اشان مرا یاری نمودند. بر خود لازم میدانم که از استاد ارجمند **آقای دکتر اشکان سامی** به خاطر زحمات فراوان ایشان در راه به ثمر رساندن این تحقیق تشکر و قدردانی نمایم. همچنین از آقایان **دکتر محمدهادی صدرالدینی** و **دکتر غلامحسین دستغیبی فرد** به خاطر راهنمایی‌های ارزشمندشان در تدوین این پایان نامه تشکر می‌کنم.

چکیده

ارائه روشی نوین برای خطازدایی خودکار نرم‌افزار

به وسیله‌ی:

حسن طنابی

ترمیم خودکار برنامه‌ها هدفی دیرینه در مهندسی نرم‌افزار می‌باشد. درحال حاضر خطازدایی فرایندی دستی، دشوار و زمان‌بر می‌باشد. رویکردهای ارائه شده در این زمینه کیفیت و قابلیت اجرایی بالا و قابلیت استفاده به صورت عملی برای توسعه‌دهندگان ندارند. در این رساله رویکردی ارائه گردیده تا بتواند از تلاش‌های قبلی توسعه‌دهندگان برای خطازدایی کدهای مشابه بهره‌برداری کند. بدین منظور پایگاه داده‌ای از خطاهای قبلی که خطازدایی گشته‌اند به همراه راهکارهای ارائه شده برای خطازدایی آنها تهیه کردیم، سپس روشی کارا به منظور جستجوی سریع در این پایگاه داده ارائه دادیم تا با هزینه کم، بتوان به سرعت کدهای مشابه با خطای جدید را پیدا نمود. در نهایت پیشنهادهایی که برای کدهای مشابه وجود دارد را به عنوان راهکار پیشنهادی به کاربر ارائه می‌دهیم.

نقطه کلیدی رویکرد ارائه شده، پایگاه داده آن می‌باشد که می‌تواند نقطه قدرت و یا برعکس نقطه ضعف آن را تشکیل دهد. در صورت داشتن پایگاه داده‌ای کامل از خطاهای مختلف، از پروژه‌های مختلف، از تیم‌های توسعه‌دهنده مختلف، می‌توان به کارایی این رویکرد اطمینان داشت. در غیر این صورت پیشنهادهای مناسبی برای خطاهای جدید یافت نمی‌شود. نقطه قوت دیگر این رویکرد زمان اجرایی کم آن می‌باشد.

واژگان کلیدی: مهندسی نرم‌افزار، خطا، خطازدایی، خطازدایی خودکار

فهرست مطالب

عنوان **صفحه**

فهرست جداول

عنوان **صفحه**

فهرست شکل‌ها

عنوان **صفحه**

فصل اول

مقدمه

فصل ۱: مقدمه

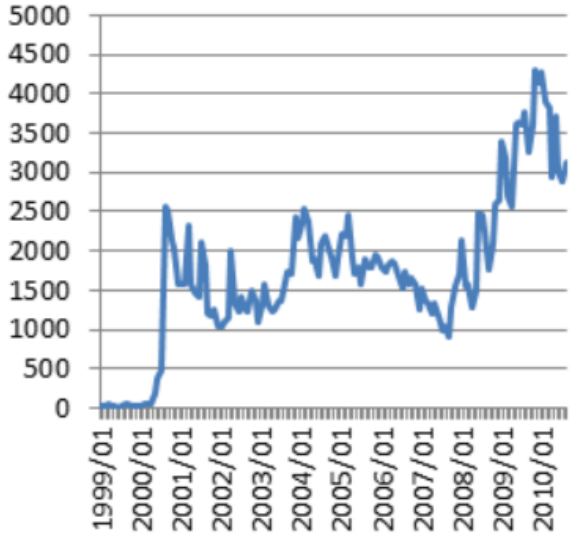
۱-۱- مقدمه

یکی از چالش بر انگیزترین موضوعات مطرح در تضمین کیفیت[1]، در شرکت‌های سازنده نرم افزار، موضوع رفع خطاهای نرم افزار است. خطاهای نرم افزاری می‌توانند در زمان پیش و یا پس از انتشار[2] نرم افزار تشخیص داده شوند. اما منابعی که می‌توان برای تشخیص و تصحیح خطاها در نظر گرفت محدود است [1]. خطاها را می‌توان به دو دسته کلی تقسیم کرد. خطاهای نحوی[3] و خطاهای مفهومی[4]. با توجه به‌این که ابزارهای خودکار بسیار قدرتمندی برای تشخیص خطاهای نحوی وجود دارند، احتمال این که خطایی ازاین دست تا زمان انتشار تشخیص داده نشود، بسیار کم است. خطاهای مفهومی به آن دسته از خطاها اشاره دارد که در اثر مشکلاتی جدای از اشتباهات نحوی و خطاهای انسانی ملموس اتفاق می‌افتند و معمولاً در اثر عدم هماهنگی در بخش‌های مختلف کد و گاهی به صورت بسیار ناملموس به وجود می‌آیند که در اینجا به سادگی نمی‌توان با بررسی کد، این‌گونه خطاها را تشخیص داد. بنا بر این در مورد خطاهای مفهومی داستان فرق می‌کند چرا که عوامل بسیار زیادی می‌توانند در بروزاین‌گونه از خطاها دخیل باشند [2].

تحقیقات نشان می‌دهد ۹۰٪ کل هزینه نرم‌افزار مربوط به نگهداری آن می‌باشد [3]. تغییر کد، تست کردن، اصلاح خطا و ... بخش‌های مهم این هزینه‌ها هستند. همچنین تصحیح خطا هم اکنون بصورت دستی توسط برنامه‌نویس انجام می‌شود و فرایند خودکاری برای انجام این کار وجود ندارد. علاوه بر این، در تحقیقات انجام شده قبلی از تجربیات پروژه‌ها و برنامه‌های با ساختار مشابه استفاده نشده است [1] [4]. تصحیح خطا در نرم‌افزارها به صورت دستی بسیار هزینه‌بر می‌باشد و هنوز روش‌های خودکاری که بصورت عملیاتی قابل پیاده‌سازی باشند وجود

1 Quality Assurance
2 Release
3 Syntax Errors
4 Semantic Defects

ندارند، از سوی دیگر تحقیقات انجام شده نتایج مناسب و خوبی با دقت کافی ارائه نداده‌اند [1] [4] [5]. شکل ۱- ۱-۱ گزارشی از تعداد خطاهای گزارش شده در KDE را تا سال ۲۰۱۰ نشان می‌دهد. در این پایگاه در ماه اول سال ۲۰۱۰ حدود ۳۰۰۰ خطا گزارش شده است و این مسئله نشان می‌دهد که کیفیت نرم‌افزار کماکان مسئله‌ای کلیدی در تولید نرم‌افزار می‌باشد.



شکل ۱- ۱-۱ گزارشی از تعداد خطاهای گزارش شده در KDE [6]

۱-۲- خطازدایی چیست؟

خطازدایی نرم‌افزار، فرایندی است که به وسیله آن توسعه‌دهندگان تلاش می‌کنند خطاهای مربوط به کد را از یک برنامه برطرف کنند. به طور معمول در حدود ۶۰ تا ۷۰ درصد زمان تولید نرم‌افزار را فاز خطازدایی تشکیل می‌دهد. در واقع، مشکلات و عدم قطعیت بسیاری در فاز خطازدایی نرم‌افزار وجود دارد.

این مسئله به این دلیل است که در هر مرحله از فرایند شناسایی خطا، مشخص کردن زمان مورد نیاز برای شناسایی و ترمیم آن امکان‌پذیر نمی‌باشد و نمی‌توان تعیین کرد که آیا این خطا کاملاً برطرف شده است یا خیر. به منظور از بین بردن خطاها در نرم‌افزار، توسعه-دهندگان ابتدا باید وجود خطا را تشخیص دهند، سپس آن را دسته بندی نمایند، محل حادث

شدن واقعی آن را در کد مشخص کنند و نهایتاً راه حلی برای برطرف کردن آن ارائه دهند (بدون معرفی کردن خطای جدید!). بعضی از خطاها بسیار وحشتناک هستند، طوری که چندین ماه و حتی در برخی موارد بدتر، چندین سال. وقت توسعه‌دهندگان را صرف برطرف کردن آن می‌کنند. توسعه‌دهندگان در فکر پیدا کردن راه‌هایی برای بهبود فرایند خطازدایی هستند. همزمان از تکنیک‌های خودکار برای کشف خطا بهره می‌برند.

در طول سالیان اخیر تکنیک‌های خطازدایی بهبود قابل توجهی داشته‌اند [5] [1][7] و در آینده نیز این روند ادامه پیدا خواهد کرد. در ادامه روند تکامل این فرایند را بررسی خواهیم کرد.

۱-۳- سیر تکاملی خطازدایی در نرم‌افزار

مطالعه تکنیک‌های خطازدایی گرایش بسیار جذابی می‌باشد. بیشتر ابداعات خطازدایی در مورد کاهش دخالت و تعامل انسان در این فرایند می‌باشد تکنولوژی خطازدایی در چند مرحله توسعه یافته است[8].

۱-۳-۱- گام اول: عصر حجر

در ابتدای زمان تولید کامپیوتر، تولید خروجی در مورد برنامه‌ی ساخته شده برای توسعه-دهندگان بسیار دشوار بود. توسعه‌دهندگان باید راه‌های جدیدی برای یافتن اطلاعات در مورد برنامه‌ی در حال اجرا ابداع می‌کردند. آنها تنها مجبور به رفع خطا نبوداند، بلکه علاوه بر آن می‌بایست ابزاری تولید می‌کردند که خطا را نیز پیدا کنند. ابزارهای سخت‌افزاری به منظور خطازدایی جز اولین تکنیک‌های مورد استفاده در این زمینه بوده‌اند.

۱-۳-۲- گام دوم: عصر برنز: استفاده از دستورات چاپ

سرانجام توسعه‌دهنگان استفاده از قرار دادن دستورات چاپ در برنامه به منظور تشخیص خطاها را آغاز کردند. با انجام این کار، توسعه‌دهندگان قادر به پیگیری مسیر اجرای برنامه و مقادیر متغیرهای مهم می‌باشند. این امر برنامه‌نویس را از فرایند زمان‌گیرساخت ابزار سخت-افزاری خطازدایی بی‌نیاز می‌کند. این تکنیک کماکان به عنوان تکنیک متداول در برخی موارد استفاده می‌شود.

۱-۳-۳- گام سوم: دوره میانسالی: خطازداهای زمان اجرا

اگرچه دستورات چاپ بهبود خوبی در تکنیک‌های خطازدایی ایجاد کردند، اما هنوز زمان و تلاش زیادی توسط توسعه‌دهندگان نیاز دارد. آنچه که توسعه‌دهندگان نیاز دارند ابزاری می-باشد که بتواند یک دستور را در واحد زمان اجرا کند و مقادیر هر متغیر را در برنامه نمایش دهد. این امر برنامه‌نویس را از صرف کردن وقت زیادی برای یافتن محل مناسب برای قرار دادن دستورات چاپ بی‌نیاز می‌کند. اینگونه بود که خطازداهای زمان اجرا متولد شدند. در واقع، خطازداهای زمان اجرا چیزی جز دستورات چاپ خودکا رنیستند. این رویکرد پیگیری روند اجرای مسیر برنامه و متغیرها بدون قرار دادن دستورات چاپ در برنامه را برای توسعه‌دهندگان مهیا می‌کند. امروزه تمام کامپایلرهای موجود در بازار همراه خود یک خطازدای زمان اجرا دارند.

۱-۳-۴- گام چهارم: حال حاضر: خطازداهای خودکار

خطازداهای زمان اجرا پیدا کردن خطا را در برنامه بسیار آسان کرده‌اند، اما قادر به تشخیص دلیل خطاها نیستند. برنامه‌نویس نیاز به ابزار بهتری برای پیدا کردن و تصحیح

خطاهای نرم‌افزار دارد. توسعه‌دهندگان دریافتند که برخی دسته‌های خطا مانند کمبود حافظه، دستری به حافظه غیرمجاز و ... می‌توانند به صورت خودکار تشخیص داده شوند. این امر گامی رو به جلو برای تکنیک‌های خطازدایی خودکار نرم‌افزار بود، زیرا فرایند پیدا کردن خطا را خودکار کرده است. این ابزار می‌تواند خطا را به توسعه‌دهنده یادآوری کند و کار او تنها برطرف کردن این خطا می‌باشد.

خطازداهای خودکار به چند صورت طراحی شده‌اند. ساده‌ترین آنها تنها کتابخانه‌ای از توابع است که می‌تواند به یک برنامه متصل شود. هنگامی که برنامه اجرا می‌شود و این توابع فراخوانی می‌شوند، خطازدا در پی یافتن خطاهای مربوط به حافظه می‌باشد و آن را گزارش می‌دهد. ضعف چنین ابزاری عدم توانایی در کشف نقطه‌ی واقعی‌ای که این خطای حافظه رخ داده است می‌باشد. این مسئله به این دلیل رخ می‌دهد که خطازدا تمام دستوراتی که برنامه اجرا می‌کند را مانیتور نمی‌کند و تنها قادر به تشخیص تعداد کمی از خطاها می‌باشد.

تمامی این روش‌ها یک اشکال عمومی دارد. تمامی آنها نیازمند حضور توسعه‌دهندگان در تمام مراحل بعد از کامپایل می‌باشد. در واقع فرایند خطازدایی از ابتدا تا به حال تغییر عمده‌ای نکرده است. ابتدا شما کد را می‌نویسید و سپس به دنبال خطاها می‌گردید. این فرایند دو مرحله‌ای کماکان وجود دارد اما تنها در سطح بالاتر. این فرایند نیازمند ادغام شدن در یک مرحله می‌باشد.

۱-۳-۵- گام پنجم: آینده نزدیک: ادغام کامپایلر و خطازداهای زمان اجرا

گام منطقی بعدی در توسعه تکنیک‌های خطازدایی خودکار، ادغام این تکنولوژی‌ها با کامپایلرها است. مشکل بزرگ در خطازداهای زمان اجرا این است که این ابزار به اندازه کافی استفاده نمی‌شود. برنامه‌نویسان کد خود را توسعه می‌دهند و هنگامی که به انتهای فرایند توسعه می‌رسند ابزار کشف خطای خود را اجرا می‌کنند. بدین ترتیب توسعه‌دهنده قادر است خطاهای برنامه جدید خود را در اجراهای اولیه کشف کند.

علاوه بر وظیفه‌مندی و استفاده آسان، ادغام ابزار خطازدای زمان اجرا با کامپایلرها، پیشرفت تکنولوژیکی قابل توجهی خواهد بود. ابزارهای مبتنی بر SCI پیشگامان این مرحله می‌باشند. این ابزار بسیار مشابه کامپایلرها هستند. ابزارهای مبتنی بر این تکنولوژی برنامه را پارس می‌کنند، درخت پارس را ایجاد می‌کنند و کد منبع نوشته شده را به کامپایلر ارسال می‌کنند. هنگامی که این ابزارها با کامپایلرها ادغام می‌شوند، فرایند خطازدایی بسیار کوتاه می‌شود. بعد از ایجاد درخت پارس، درخت می‌تواند به منظور ارزیابی به ابزارهای خطازدایی ارسال شود.

۱-۴- جمع‌بندی

در این فصل اهمیت خطازدایی، مشکلات و سیر تکاملی آن را بیان کردیم. خطازدایی نرم‌افزار، فرایندی است که به وسیله آن توسعه‌دهندگان تلاش می‌کنند خطاهای مربوط به کد را از یک برنامه برطرف کنند. به طور معمول در حدود ۶۰ تا ۷۰ درصد زمان تولید نرم‌افزار را فاز خطازدایی تشکیل می‌دهد. سپس روند پیشرفت این تکنولوژی را در پنج دوره مختلف بیان کردیم: ۱) خطایاب‌های سخت‌افزاری ۲) استفاده از دستورات چاپ ۳) خطازداهای زمان اجرا ۴) خطازداهای خودکار ۵) ادغام کامپایلر و خطازداهای زمان اجرا.

در فصل بعدی به دلیل اهمیت و وابستگی مسئله خطایابی ابتدا کارهای انجام شده در این زمینه را بیان می‌کنیم. سپس به بررسی کارهای انجام شده در زمینه خطازدایی می‌پردازیم.

فصل دوم

پیشینه تحقیق

فصل ۲: پیشینه تحقیق

۲-۱- مقدمه

در این فصل کارهای انجام شده در زمینه خطازدایی را بررسی و نقاط ضعف و قوت هرکدام را بیان می‌کنیم. خطازدایی خودکار در نرم‌افزار سابقه دیرینه‌ای ندارد و تحقیقات انجام شده در این زمینه خواسته‌های توسعه‌دهندگان را برآورده نکرده است. کارهای گذشته بیشتر در زمینه کشف و پیش‌بینی خطا بوده است و خطازدایی خودکار در سال‌های اخیر مورد توجه قرار گرفته است.

مطالب این فصل را به دو بخش کلی تقسیم می‌کنیم. به دلیل اهمیت و وابستگی خطایابی در نرم‌افزار ابتدا کارهای مهم و خوب در زمینه پیش‌بینی خطا را بیان کرده و سپس به بررسی کارهای انجام شده در زمینه خطازدایی خودکار می‌پردازیم.

۲-۲- پیش‌بینی خطا

تلاش‌های گذشته با هدف پیش‌بینی خطا را در سه گروه مورد بررسی قرار می‌دهیم: معیارهای نرم‌افزار[1]، معیارهای وابستگی[2] و معیارهای تاریخی[3]. ابتدا به بیان توضیحی در مورد شیوه کار کردن این معیارها می‌پردازیم.

۲-۲-۱- معیارهای نرم‌افزار:

1 Software Metrics
2 Dependency Metrics
3 Historical Metrics

معیارهای نرم‌افزار برای اندازه‌گیری درجه پیشرفته بودن یک محصول و یا یک فرآیند نرم افزاری به کار می‌روند. معیارهای نرم‌افزاری به چندین گروه تقسیم می‌شوند: معیارهای محصول[۱]، معیارهای فرآیند[۲]، معیارهای پروژه[۳] و معیارهای منبع[۴]. معیارهای نرم‌افزاری که در پیش‌بینی خطا به کار می‌روند معیارهای محصول می‌باشند که از مشخصات کد سیستم نرم افزاری استخراج می‌شوند. این معیارها به سه گروه تقسیم می‌شوند: معیارهای اندازه[۵]، معیارهای پیچیدگی و معیارهای کیفیت [۶] [2] . معیارهای اندازه بر اساس تعداد خطوط کد برنامه محاسبه می‌شوند مانند تعداد کل خطوط برنامه[۷]، تعداد خطوط توضیحات و ... معیارهای میزان نگهداشت‌پذیری[۸] و قابلیت تست برنامه وابسته است از جمله معروف‌ترین معیارهای پیچیدگی معیارهای پیچیدگی مک کیب[۹] و معیارهای هالستد[۱۰] می‌باشند. معیارهای مک‌کیب پیچیدگی کد را بر اساس تعداد مسیرهای کنترلی محاسبه می‌نماید[3] . هالستد معیارهای خود را بر اساس ارتباطات ریاضی بین اجزای کد، پیچیدگی کد و نوع زبان برنامه نویسی مطرح کرد[4] . معیارهای اتصال[۱۱] و پیوستگی[۱۲] از معروف‌ترین معیارهای کیفیت می‌باشند که بالا و یا پایین بودن اندازه این دو معیار نشان دهنده کیفیت محصول و یا فرآیند نرم‌افزاری است[5] . معیارهای کیفیت معیارهایی می‌باشند که درجه آن‌ها می‌تواند تولیدکنندگان نرم‌افزار را در مورد توانایی دست کار کردن سیستم‌شان مطمئن سازد.

۲-۲-۲- معیارهای وابستگی:

4 Product Metrics
5 Process Metrics
1 Project Metrics
2 Resource Metrics
3 Size Metrics
4 Quality Metrics
5 Total Line of Code
6 Maintainability
7 McCabe metrics
8 Halstead metrics
9 Coupling
10 Cohesion